Volume 08 Issue 01 Published, February 18, 2004 ISSN 1535-864X

Intel® Technology Journal

Intel® Pentium® 4 Processor on 90nm Technology

Full Hold-Scan Systems in Microprocessors: Cost/Benefit Analysis

A compiled version of all papers from this issue of the Intel Technology Journal can be found at:
http://developer.intel.com/technology/itj/index.htm

Full Hold-Scan Systems in Microprocessors: Cost/Benefit Analysis

Ravishankar Kuppuswamy, Technology and Manufacturing Group, Intel Corporation
Peter DesRosier, Technology and Manufacturing Group, Intel Corporation
Derek Feltham, Desktop Platforms Group, Intel Corporation
Rehan Sheikh, Desktop Platforms Group, Intel Corporation
Paul Thadikaran, Desktop Platforms Group, Intel Corporation

Index words: Microprocessor, Scan, Test, DFT, DFM, ATPG, DPM, Fault Grading

ABSTRACT

Ever-shrinking microprocessor product development times require enhanced High-Volume Manufacturing (HVM) techniques. This paper describes the full hold-scan testing system implemented in the 90nm Intel® Pentium® 4 processor. Benefits of this scan system include significantly reduced functional test-writing and fault-grade effort, extensive initialization of the design for test and debug, massive visibility into the design for post-silicon debug and fault isolation, and ultimately, a significantly accelerated ramp to production test quality. Any full hold-scan system such as this impacts timing, power, area, and schedule. In a high-performance microprocessor, in particular, this significantly impacts product viability and must be closely managed. In this paper, the Intel full hold-scan system is described, particularly the design challenges, cost optimizations, and test benefits, and we also discuss the costs and benefits of having implemented this successful testing system.

INTRODUCTION

Our full hold-scan system (Figure 1) comprises a full-chip scan bus that acts as a communication channel between the Test Access Port (TAP) and all units. The TAP controller converts data between the TAP clock domain at the pins and core clock domains at the full-chip scan bus interface. This controller also supports serial (TDI/TDO) and parallel modes (36 in/36 out data & address bus pins). There are 29 Scan Functional Units (SFUs) distributed across the chip that interface between the full-chip scan bus and the intra-unit scan chains. Each SFU supports 18 logic chains, 5 reserved chains and up to 3 custom chains. The logic chains provide access to ~200k scan sites in the design. The scan sites are implemented with hold-scan cells, which provide a full shadow of the machine state and enable non-intrusive operation while the system is running or while system clocks are frozen.

Full hold-scan systems have an inherent die area cost and have a critical impact on the timing performance and power metrics of the microprocessor. The transistor scaling for the development of high-speed circuits further exacerbates the power problem. Additionally, the scan circuits contribute to local device and wire congestion leading to design convergence issues. We used a variety of architectural and circuit design solutions to build a commercially viable low-power full-scan system. We describe the full-scan circuits in the latest 90nm Intel Pentium 4 microprocessor. We also discuss the various design trade-offs (cost vs. benefit) and the evolution of Intel's first full hold-scan system design in a microprocessor.

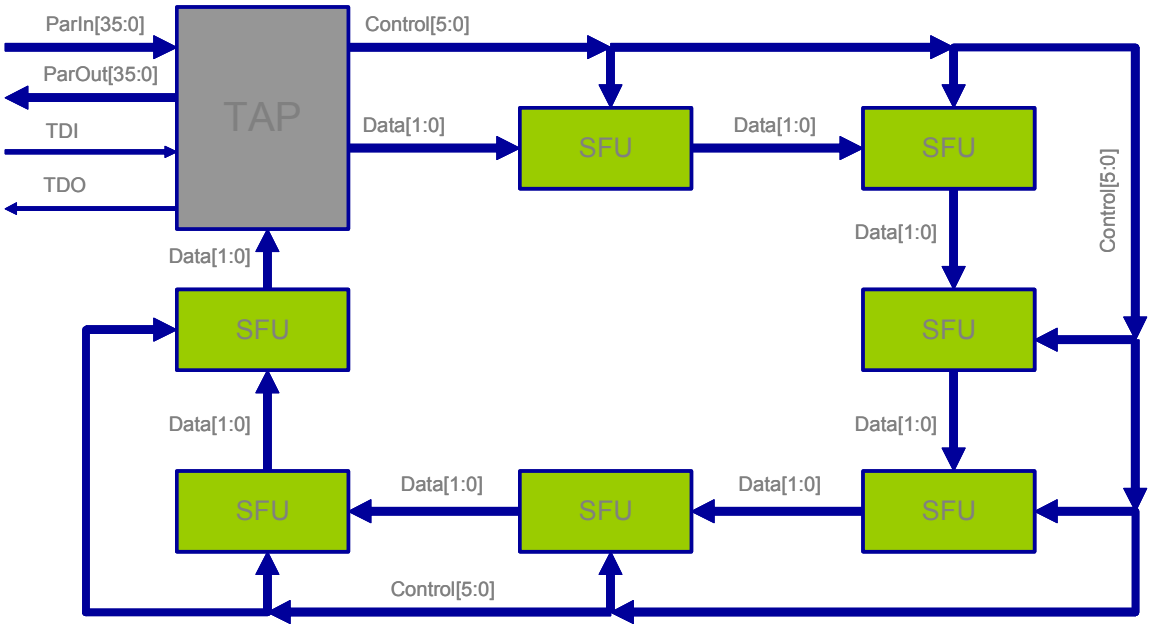

Figure 1: Full-chip scan system

DESIGN AND TEST CHALLENGES

Intel's move to full hold-scan in this design generation was motivated fundamentally by the need to continue accelerating new product development. Historically, Intel has used custom Design For Test (DFT) features and post-silicon test-writing to create high-quality production test suites for lead microprocessor designs [1]. Past Intel® processor design teams have consumed upwards of 50 person-years to develop the production test suite for a new design. The significant post-silicon effort of this approach was justified by the product cost savings that could be realized through saving the die area and timing margin that more systematic DFT would have cost. This cost/benefit trade-off has remained true for many generations of Intel Pentium® processor design, as driven primarily by the economics of extremely HVM.

With every design generation, however, the challenge of creating a high-quality production test suite [2] in a timely fashion with reasonable resources is increased. And in every design generation, the business environment changes to even further increase the economic emphasis on time-to-market and on the need to move critical design resources to the next project as soon as possible. In the latest microprocessor design generation, all these factors finally led us to doing extensive systematic DFT in our lead processor.

The benefits that would have to be realized to justify this move to full hold-scan were as follows:

1. Reduced test generation and fault grade effort.
2. Increased visibility into internal state, for post-silicon debug and fault isolation productivity.
3. Improved initialization for maximally effective functional signature testing.
4. High burn-in toggle capability in a limited pin-count test environment.
5. Significantly accelerated ramp to HVM, enabling production test quality.

Automated DFT provides the promise of quick, reliable test generation for a new product and all of its design proliferations, quick stabilization in factory test programs during the critical debug and ramp phase, and ultimately significant factory cost savings because of a much earlier reduction in system-testing to ensure outgoing quality levels.

Reduced debug time is a fundamental goal in itself. It reduces the overall time-to-market of a product and also accelerates the pace with which design modifications can be turned for debug onion-peeling. The standard interface and massive observability of a scan system is understood industry-wide for its many benefits to debug.

® Intel and Pentium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Initialization is also a key engineering challenge to rapidly achieving high test quality, particularly for a signature-based functional test of a design. The inclusion of a full hold-scan system and minimal extra DFT provides a much easier path to massive system initialization in advance of a deterministic test–both for functional signature testing and for Automatic Test Pattern Generation (ATPG).

Finally, maximizing toggle coverage (circuit excitation) during burn-in on a low pin-count test board can be achieved through the use of full hold-scan.

All of the above led to a capability to accelerate the ramp from initial tape-out of a new product to high-volume, high-quality stable manufacturing, and also to maximizing the capability of the manufactured product.

Even given the expected test benefits, it was still critical to pay attention to the historical design challenges that had worked against choosing full hold-scan in the past. To ensure that scan achieved the right economic trade-off for the product, the following design goals also had to be met:

1. Integrate debug and test capabilities into a single cost-optimized scan system.
2. Minimize scan impact to die area, power, timing, and design schedule.
3. Minimize reliability impact due to contention under test.
4. Keep the overall system, library requirements, and design rules as simple as possible, so as to minimize validation complexity and maximize achievable coverage benefit.

These design requirements led to the scan system described in this paper. In particular, requirements 3 and 4 led to the specific full hold-scan system that was adopted. Specifically, we tried to minimize variation in our scan system design, so as to streamline the effort for all peripheral teams involved in getting full hold-scan into the Intel IA-32 microprocessor design environment for the first time, and to maximize the chances of taping out a fully bug-free scan system into first silicon. Other technical requirements that shaped the system we adopted included the need for compatibility with the structural testers [3][4] in use in our high-volume test floors, enabling efficient parallel access for HVM test time minimization, providing serial access for burn-in and system debug, providing hooks for making easy enabling/disabling changes to manufacturing test programs, and providing extensive signature capability for maximum test quality from our extensive legacy functional test base used for at-speed testing.

SCAN SYSTEM ARCHITECTURE

Scan Functional Unit Features

The full hold-scan system comprises 29 SFUs distributed across the different functional blocks of the microprocessor. Each SFU is an intermediate control station in the scan system that can be configured to access different scan chains in the corresponding partition. SFUs are configured using address scan registers that exist in each SFU and get connected as a serial shift chain in scan address mode (Figure 2). Each SFU provides access to 5 reserve and 3 custom chains as shown in Figure. 3. Custom chains are available for additional future visibility. Reserve chains are used for various scan usage modes (which are mentioned later in this paper). We do not, however, present details of reserve chain usages in this paper.

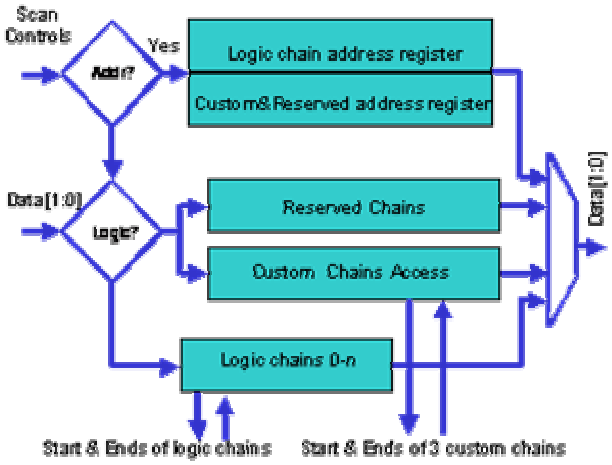

Figure 2: Scan functional unit block diagram

Logic Scan Chains

There are ~200k scanned states on the die. The scan system for 15% of these states operates at full-core frequency. These full-core frequency scanned states are consolidated onto 2 of the sub-chains within each SFU. These chains are referred to as “fast” chains (total fast chains = 70). The remaining 16 sub-chains in each SFU operate at (up to) one-half of the core frequency. These SFU sub-chains are referred to as “slow” chains. All 18 chains in each SFU share the same clocking network. Originally all chains were designed to run at the full core frequency. This approach provided additional benefits such as running “transition fault” tests at the core frequency. However the scan logic on 16 of the chains in each SFU was re-designed specifically to save power. The two “fast” chains in each SFU were preserved for

usage in the full frequency modes (snapshot and signature modes).

TAP Controller and Full-Chip Scan Bus

The Test Access Port (TAP) controller converts data between the TAP clock frequency domain at the pins and internal core clock domain at the full-chip scan bus interface. The scan system in the TAP operates in either serial or parallel mode. In parallel mode, the TAP communicates with an Automated Test Equipment (ATE) tester using 36 input and 36 output pins. In serial mode, the TAP communicates with the ATE tester using TDI and TDO. The TAP always shifts data into the two scan bus data bits simultaneously. The full-chip scan bus is comprised of two bits of data and five bits of control signals. The two bits of data serially connect each of the SFUs while the 5 bits of control signals are routed to each of the SFUs as shown in Figure 4 in a balanced tree with equal latency. The scan bus provides a non-intrusive operation while the system is running or while the system clocks are frozen. In parallel mode, the tester sends out data (36 bits every TAP clock cycle) on the input bus. Eighteen of the inputs received by the TAP are shifted into one of the data bits in the scan system bus while the other 18 inputs are shifted into the other data bit of the scan system bus.

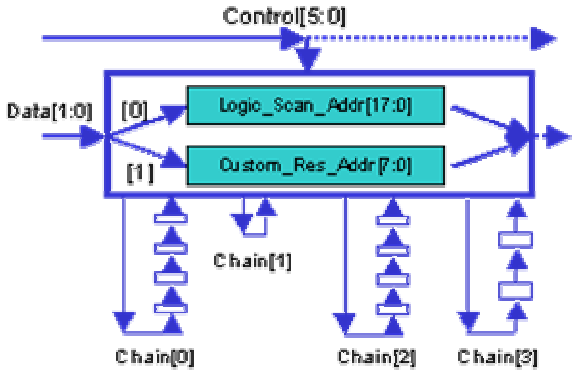

Figure 3: Addressing inside the SFU

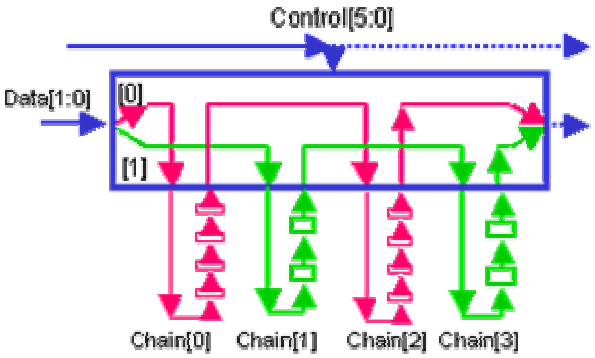

Figure 4: Logic chain connections inside the SFU

HIGH-LEVEL OPERATIONS AND MODES

There are two primary arenas in which the scan system is utilized. HVM, and Silicon Debug (1st Silicon Debug, System Debug, Low Yield Analysis Debug (LYA)).

Snapshot for 1st Si System and LYA Debug

In both the tester and the system debug environments, system data are captured in the scan states "on the fly" while the system clock is running; this is called a "*scan snapshot.*" These data are then shifted along the scan chains and out of the chip in an operation called a "*scan dump.*" At the chip interface, the captured states can be serially shifted out on TDO pins or converted into parallel data and sent out on external data and address pins.

During system debug, the architectural state of the machine is dumped periodically using the Periodic System Management Interrupt mechanism (details are not included in this paper). When the system debugger would like to know the "scannable" system state, he or she can also perform periodic scan dumps. Consequently, the scan dump operation must be non-destructive (not change the state of the machine), and it must be possible to continue normal system operation during and following the scan dump.

A common use of scan snapshot on an Automated Test Equipment (ATE) tester is to run a given test, perform a scan snapshot at cycle N and dump the scan data, re-start the test, do a scan snapshot at cycle N+1 and dump the scan data. Continue in this fashion until K cycles of snapshot data have been collected.

Snapshot Compression for 1st Si Debug

For repeatable errors, data are dumped and compared several times to find when data goes from good to bad. To reduce the amount of data that need to be collected, scan chain data are compressed on-chip and only a "signature" is dumped out. The scan chain ("fast" chains only) outputs are fed through Linear Feedback Shift Registers (LFSR) in each SFU and finally compressed further by another LFSR in the TAP to produce the signature. About 35k bits of snapshot data are compressed down to 32 bits.

On-Die Snapshot Diff for 1st Si Debug

Snapshot compression is more useful as part of a technique called "*on-die snapshot diff.*" Often the debugger is only looking for differences in the data between a given test run and a known good run, thus the snapshot data are compressed on-chip. The known good signature can also be stored on-chip. The signatures produced by subsequent test runs can be compared with

the golden signature on-chip, with only a pass-fail bit shifted out of the chip.

Signature Mode for Functional Tests in HVM

The shortcoming of the previous usage models is that, to detect an error, the debugger must first find a point in time during the test run where the error appears when a snapshot was taken. "*Signature mode"* solves this problem by compressing scan snapshot data taken every clock cycle over a period of time. In addition to using the LFSRs mentioned above to compress the scan data spatially, the individual scan cells can compress the scan data temporally. For example, instead of overwriting an old bit of snapshot data, scan cells can XOR the old data with new bits of snapshot data.

Signature mode is primarily useful in the production test environment; especially while running hand-written architectural tests. Instead of having to understand the full-chip architecture to propagate a fault effect all the way to a chip pin; the test writer's job is greatly simplified: he or she only needs to figure out how to propagate the effect to a signature scan node.

Automatic Test Pattern Generation for Various Faults Types in High-Volume Manufacturing

The previous usage models only require the scan cell to be able to observe a system state, which is reached by running an architectural test. Scan is a much more powerful feature when it is used to control, or modify, the system state. Then, a stimulus pattern can be shifted into the scan chain and applied to the system logic. The response pattern can then be shifted out of the scan chain and compared with an expected response obtained through simulation. An ATPG tool determines the stimulus and response patterns.

At the time of this design a microprocessor-sized netlist was too large for an ATPG tool to handle all at once, thus the design was partitioned into multiple clock domains. Scan tests were generated and applied to one domain at a time. The simplest type of fault for which to test using ATPG is the stuck-at fault.

Toggle Coverage in Burn-in Mode in High-Volume Manufacturing

During burn-in testing, the scan system is used similarly to how it is used in ATPG mode, except that the serial interface to the scan system (via only the TAP port) is used instead. Content loaded into the design for burn-in testing can be either pseudo-random or generated by the ATPG tools.

Device Initialization in Test Modes in High-Volume Manufacturing

The scan chains are also used for device initialization in test modes. In this usage case, the scan system is loaded with all 0s during reset pin assertion, and then a scan store operation is executed to initialize the state of 200k non-array sequential elements in the design to 0. This is done early in the microprocessor reset sequence.

FULL SCAN DESIGN EXECUTION

Floor-Planning

The full hold-scan execution effort started with a detailed floor-planning exercise to accommodate the 200K scanned states on the chip. A detailed scan device area estimation helped partition the full-chip functional block boundaries as shown in Figure 5. We identified key congestion areas to determine the critical cut of the die and help shape the boundaries. Regular audits of bottom-up area estimates were used to alter floor-plan assumptions. The clock and control signal distribution networks account for ~35% of the scan device area and hence required congestion studies. The control signals were routed in a balanced tree network to the different SFUs and scan elements to optimize the signal tracks.

Figure 5: Full-chip floor plan

Library Development

The library cells are the fundamental building blocks of the full hold-scan system. We designed independent scan libraries to cater to the two distinct design methodologies on the die: Cell-Based Design (CBD) and Embedded Building Blocks (EBB). The library cells are designed independently and pre-characterized to describe their

behavior for timing, logic, noise, reliability, etc. The microprocessor uses a wide variety of latches and flip-flop circuits to implement the logic functionality. The scan system now requires scanned variants of every latch and flip-flop on the die. A scanned latch shown in Figure 6 comprises two distinct circuits: a system latch and a scan gadget. The system latch is the pristine storage element catering to the system functionality needs, while the scan gadget is comprised of a fixed size storage element and interface circuits to meet communication needs (shift, capture, store) as dictated by the scan architecture. The interface circuits are sized in relation to the system latch.

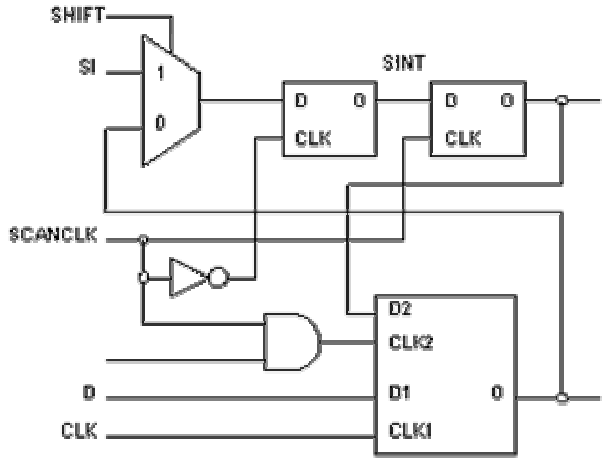

Figure 6: System latch with scan gadget cell

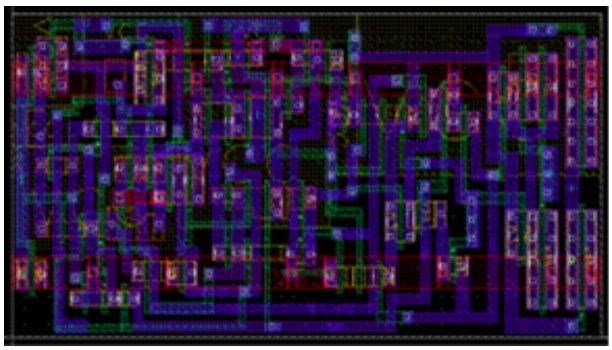

Figure 7: System latch with fast scan gadget layout

In a full hold-scan design, the large usage numbers of scanned state elements (~200K) impose a high standard on the scan gadget design. Individual scan gadgets have been designed for use in fast and slow chains. The fast gadget shown in Figure 7 needs to support shift, snap-shot and store operations at full-speed, while the slow one needs to operate the same functions at half core-speed. This not only provides an opportunity to size down the devices but also allows 100% use of low-leakage devices in slow gadgets. Special care was taken to reduce the amount of clock switching capacitance, a source of dynamic power. Library cells were characterized first at the cell level to meet all scan timing requirements. Scan cell timings were designed with an additional 5-10% margin to guarantee that process variation does not push the scan circuits to be the speed-limiting paths on the die.

Scan Design Flows and Methodologies

There are two unique types of design methodologies used on the microprocessor: EBB and CBD [5]. The physical break-up of the two types of blocks is approximately equal on the chip. While EBBs are hand-crafted by individual designers, the CBD blocks use automated techniques to implement the design. Although the implementation flows of the two types of blocks are vastly different, there is little difference in the logic model. The design has a small fraction of blocks that are deviant from the CBD methodology called Structured Data Path (SDP) blocks. SDP blocks need hand-drawn schematics and some manual intervention to help in the automation of the convergence of the design.

The logic representation of a microprocessor design undergoes significant early changes during the functionality definition and logic convergence phases of the design. To enable validation of the logic and downstream steps of a microprocessor design flow, snapshots of the logic model are released on a regular basis. To enable such rapid changes in succession, it is important to keep the scan-related steps and collateral minimal while still enabling validation of scan features in the model. This requirement for a light-weight process drove the scan flow strategy. The key steps involved are scan selection, insertion, ordering and stitching, clock tree synthesis, and rules and checkers.

Scan Selection

The scan selection step involves marking the correct states to be scanned. This step is required even in a full hold-scan methodology to enable exceptions and to avoid scanning transparent latches, states supporting the scan system, and other test or debug features.

Scan sites were selected in the logic using a scan selection tool *scansel*. The *scansel* needs to not only identify transparency but also recognize the clock phase of the different states when the global clock of the microprocessor is shut off. This is a key requirement to guarantee functionality of the different modes of the scan system. To enable appropriate accounting, all states in the logic microprocessor design were divided into three classes: *scan, do_not_scan & scan_exception*. Examples of states that fall into *do_not_scan* are transparent latches, states supporting other test/debug features, etc. Examples of states that are classified as *scan_exception* are states in blocks that are exempt from full scan requirements. All blocks in the design fall into two full scan categories: a) full scan targeted or b) full scan exempt.

Scan Insertion
The scan insertion step involves incorporating scan by replacing the non-scan state elements with scan equivalent state elements and connecting the scan control ports of the scan cell to the scan control signal distribution network. In EBBs this was a manual operation while in CBD blocks, automation manipulated the netlist representation of the design. For semi-custom (SDP) and synthesized (CBD) blocks, scan insertion was automated through the use of API functions available within the logic synthesis tool *Design Compiler*[*].

Scan Ordering and Stitching
The scan ordering step involves connecting the scan-out port of a scan cell to the scan-in port of the next scan cell to form a serial chain. This enables a scan ordering that is optimal in terms of usage of chip area used to route the scan chain. This functionality was implemented using features available with the place and route tool *Apollo*[*]. The scan chain ordering flow was constrained to meet slope rules for all scan signals and full functional timing goals on select scan chain paths. To meet these requirements, large repeater networks were designed to enable distribution of scan signals. Additionally, these requirements had to be met within tight block area constraints.

Scan Clock Tree Synthesis
The scan states on the die use generic core clocks to avoid additional global clock distribution costs. Although the global clock grid is shared between system and scan, the scan local clock buffers have been specifically optimized for power and area constraints. There exists a special local scan clock network delay and slope tuned to meet scan functionality while limiting degradation to system performance. The scan clock network below the global clock grid consists of a Regional Clock Buffer (RCB) and a Local Clock Buffer (LCB). The RCB is a scan-function-enabled driver that services a small number of LCBs while meeting fan-out guidelines. The scan LCBs are un-gated buffers driving ~10-15 scan states.

Scan Rules and Checkers
The scan states are present across the die in and around different types of circuit configurations. A set of rules was developed to ensure functionality and maintain consistency across the design. They primarily belong to two classes: Scan Friendly Array (SFA) and Scan Friendly Mega Block (SFMB). The former deals with scan logic interaction with memory arrays, and the latter focuses on interactions between scan elements of one logic block with another neighboring block.

Design Rule Checkers (DRC) are deployed to check for compliance of the design to scan rules. These scan rules support the scan ATPG strategy adopted, and compliance to these rules is critical to the success of obtaining the full scope of the test coverage benefits. The result from the DRCs was a pass/fail for a scan-targeted block.

COST AND BENEFIT ANALYSIS

Timing/Performance

System performance is critically dependent on the scan gadget architecture and operating modes. The scan gadget circuits add capacitance to the system nodes thereby causing degradation to the setup and clock-to-out times of the system. The scan gadget design is tuned to work well within its operating modes to limit this timing degradation. The scan contribution of the setup and clock-to-out penalties are summed into a new metric called "Black-hole time." A select number of families were simulated and the black-hole times are reported in Figure 8. We used a variety of circuit design techniques to reduce black-hole time penalties. Furthermore, the system timing degradation will translate into actual maximum frequency (Fmax) loss if and only if the impacted system paths are the speed-limiting paths on the manufactured chip.

[*] Other brands and names are the property of their respective owners.

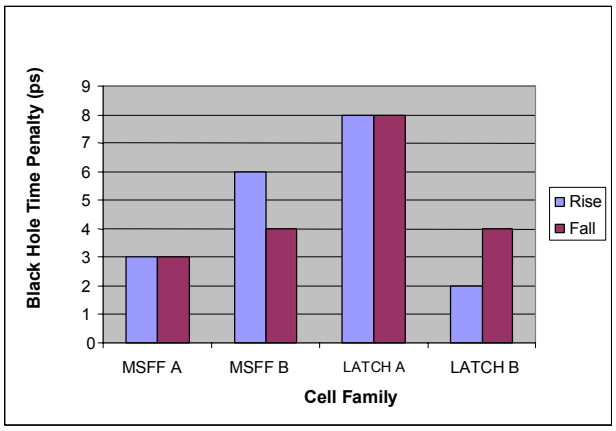

Figure 8: Black-hole time performance penalty

Die Area

The full scan system area contributions are primarily shared by the scan gadget, scan clocks, and control distribution circuits. These critical components of the scan system tend to affect the critical cut of the die. This issue can be mitigated with early floor planning and with the use of optimized place and route tools. Our scan methodology allowed for full scan exception cases in select areas to ease local device and wire congestion.

Power

Although the dynamic power is important from the stand-point of power delivery capability in test systems, the critical contribution from scan is the leakage power during normal system operation. The scan system was partitioned into fast and slow scan chains with ~15% cells in fast chains. This inherently allowed us to reduce leakage power to a minimum in slow chains. Even in the fast chains, the scan operating modes were optimized; i.e., all non-essential functions were slowed down to optimize for power. The scan gadget nodes are parked in non-toggling state, and furthermore, scan clocks do not toggle during normal operation. The control signal distributions were architected to operate at low frequency thereby saving power. Scan power has been reduced to ~2.5% of the total power for the processor.

Time to Quality and Factory Savings

The primary goal of adding full hold-scan into the design was to reduce HVM test development effort for the product, without compromising test quality. The final fault coverage for this product was achieved through a combination of scan ATPG testing, other DFT-based tests (such as directed array testing), and full-speed functional tests. The full-speed functional test base consists of significant existing "legacy" tests and many newly-written tests targeted at fault coverage enhancement. All classes of tests are graded against the current design, to assess coverage and ensure quality. Against an overall graded single-stuck-at fault coverage goal of ~96% for this product, scan provided unique coverage of 6 percentage points above all other content in the test base, as shown in Figure 9.

This unique coverage translated directly to saving significant manual test-writing effort in the "last mile" (the most labor-intensive portion) of test development, that would have otherwise had to be done. Recent Intel test development experience (from both the current high-volume 130nm design and the new 90nm Intel® Pentium® 4 microprocessor) has indicated that "last mile" manual test development costs on the order of 5 person-years effort per percentage point of full-chip coverage. Hence, full hold-scan in this design directly saved ~30 person-years of test development for HVM. Practically speaking, this can alternately be stated as a 6-month pull-in from test development schedule to quality for the product, which is a significant fraction of our typical timeline from 1st silicon through debug to product introduction.

Apart from direct engineering cost savings, the reduction in effort has an even more significant benefit in the factory: reduction in time-to-quality leads to earlier detection of real testing issues (as opposed to start-up issues), much better predictability of the test program ramp to stability, and much quicker cost-reduction of expensive screening tests, which are always employed in early testing of a new product. This alone has been estimated to represent large savings in factory test cost, in the early testing ramp for this 90nm-generation microprocessor.

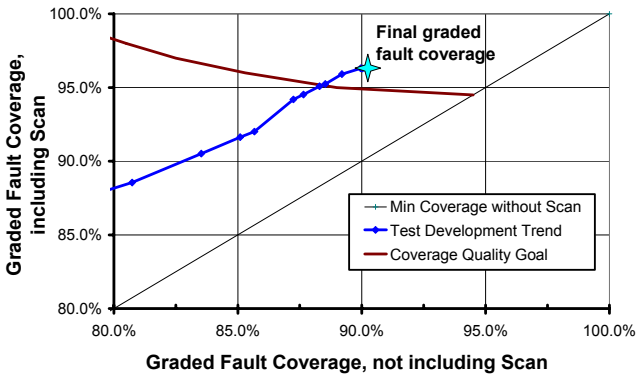

Figure 9: Fault coverage goals

Yield Learning and Quality Improvement

The extent of scan in this design also provides a powerful diagnostic technique for failure analysis of bad parts in the early days of product ramp. This has benefits both for yield learning on a new process, and for test program

improvement through early and quick identification of test holes in the random logic of the die. For many product generations, Intel has had industry-leading array DFT and techniques for failure analysis and fault isolation in array logic, but the same process for random logic has been highly customized and laborious. With the advent of full hold-scan, we bring the fault analysis time and cost in random logic down from weeks to hours.

Historically for Intel, the potential test-writing effort savings possible through full hold-scan was swamped by the die area cost in the volumes of product that we run. The potential debug and manufacturing benefits of full hold-scan were well understood, but existing methods were mature, productive, and cost-effective enough to keep HVM test development well off the critical path for new product ramp. The pressure that has been mounting, and what changed specifically with this processor generation, is the need for even quicker design cycles, and historically unprecedented manufacturing volume shift to each next-generation product. These factors shifted the economic return on investment for full hold-scan from negative to positive for this generation of product. We have taken significant cost to move our design processes to full-scan, but we are in turn reaping the benefit in the many debug and manufacturing ramp capabilities that this provides.

Additional significant benefits of a full hold-scan system are widespread initialization during the production test reset sequence, significant internal visibility into the design for debug, and burn-in toggle coverage capability. One thing that is uniquely beneficial to Intel's full hold-scan design is the ability to shift state into the design without causing any contention in the design. This is used in production test for design initialization prior to functional signature mode or ATPG testing, and in our burn-in environment for pseudo-random toggle coverage maximization without risking chip damage. The second capability, which is unique to our full hold-scan design, is the debug ability to capture almost full internal state in either the tester- or system-based debug environments, while running tests at full speed. This opens a new door for observability that is now being leveraged as we continue to develop debug capabilities around this test feature.

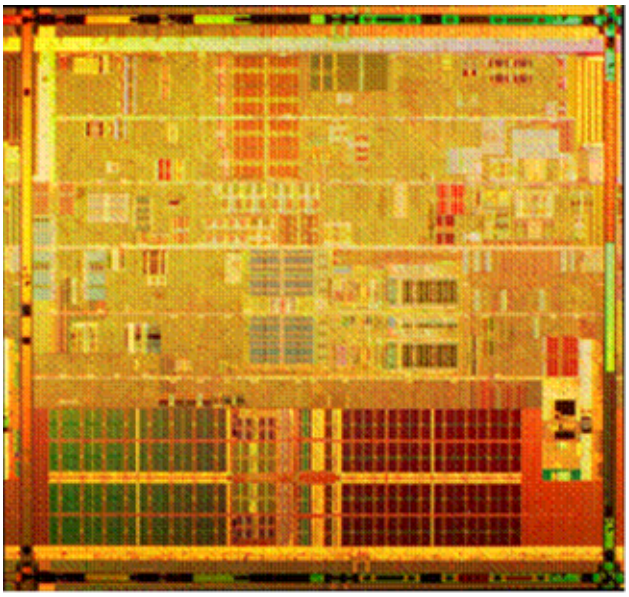

Figure 10: 90nm Intel® Pentium® 4 microprocessor die photo

CONCLUSION

Shorter product cycles are driving the need for a reduced test generation and fault-grading effort. There is an imminent need for a quick, reliable, test content and methodology that is capable of being proliferated to future generations of microprocessor design. Our full hold-scan system not only provides a solution to this problem but also helps reduce logic and speed-path debug time. One needs to pay close attention to the die area, power, and system performance impact to harness all the benefits of this new full hold-scan system. In summary, Intel's first low-power performance-friendly full hold-scan CPU has been designed and integrated into the Intel 90nm Pentium 4 microprocessor shown in Figure 10. This scan system will not only help decrease the time-to-market of this product but also set the standard for all our future generations of high-speed microprocessors.

ACKNOWLEDGMENTS

The full hold-scan design is a culmination of several contributors from design, device and test, including several key technical leaders from both the product and the technology groups at Intel. We thank Mike Mayberry, Adrian Carbine, Greg Taylor, Dave Nedwek and Jim Wilson for their help in reviewing this paper and providing data, figures, and valuable insight for this paper. The authors also thank the scores of design and product engineers, tool developers, and debuggers who helped make the full hold-scan product a reality.

REFERENCES

[1] Carbine, A. and Feltham, D., "Pentium® Pro Processor Design for Test and Debug," in *Proceedings of the 1997 IEEE International Test Conference*, 1-6 Nov. 1997, pp. 294-303.

[2] Sengupta, S. et. al., "Defect-Based Test: A Key Enabler for Successful Migration to Structural Test," in *Intel Technology Journal*, Q1 1999.

[3] Mayberry, M.; Johnson, J.; Shahriari, N., and Tripp, M., "Realizing the benefits of structural test for Intel microprocessors," in *Proceedings of the 2002 IEEE International Test Conference*, 7-10 Oct. 2002, pp. 456-463.

[4] Tripp, M., "On-die DFT based solutions are sufficient for testing multi-GHz interfaces in manufacturing (and are also key to enabling lower cost ATE platforms)," in *Proceedings of the 2002 IEEE International Test Conference*, 7-10 Oct. 2002, p. 1232.

[5] Rich, S.E., Parker, M.J. and Schwartz, J., "Reducing the frequency gap between ASIC and custom designs: a custom perspective," in *Proceedings of 2001 Design Automation Conference*, 18-22 June 2001, pp. 432-437.

AUTHORS' BIOGRAPHIES

Ravishankar Kuppuswamy is a design manager in Intel's Technology and Manufacturing Group, specializing in mixed signal design and Design for Test (DFT) areas of high-speed microprocessors. Ravi received both a B.S. degree in Electrical Engineering and a M.S. degree in Chemistry from the Birla Institute of Technology and Science, Pilani, India in 1994 and a M.S. degree in Electrical Engineering from Arizona State University in 1996. He has worked on a large number of product areas including IO, clock generation, clock distribution, and features specifically targeted for High-Volume Manufacturing (HVM). Ravi's e-mail is ravishankar.kuppuswamy at intel.com.

Peter DesRosier: is a design manager in Intel's Technology and Manufacturing Group, specializing in DFT features and architectures and production test methods for high-speed microprocessors. Peter received his B.S. degree in Electrical Engineering from Montana State University in 1985. Peter's e-mail is peter.desrosier at intel.com.

Derek Feltham is a principal engineer in Intel's Desktop Platforms Group, specializing in DFT and production test methods. His technical interests include DFT features, test development productivity, and test quality prediction methods. Derek received his B.S. and M.S. degrees in Applied Science from the University of Toronto and a Ph.D. degree from Carnegie Mellon in 1992. Derek's e-mail is derek.feltham at intel.com.

Rehan Sheikh is a senior staff engineer in Intel's Desktop Platforms Group, specializing in DFT and HVM methods. His technical interests include DFT feature design, validation, initial Si debug, and advanced test methods. Rehan received his Masters of Electrical Engineering from Georgia Tech in 1994. Rehan's e-mail is rehan.sheikh at intel.com.

Paul Thadikaran currently leads a DFT Tools team in Intel's Desktop Platforms Group focused on developing DFT tools and solutions for IA-32 CPU designs. Paul received his Ph.D. in Computer Science from SUNY-Buffalo in 1997. His technical interests are CAD tools for DFT and Test generation and CAD algorithms for design optimizations. His e-mail is paul.thadikaran at intel.com

® Pentium is a registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

For further information visit:

developer.intel.com/technology/itj/index.htm